

CLAIMS

I claim :

1. A computer system comprising at least one data source and a mapping system wherein object programming applications are tailored to delegate both the accessing of a data source and the generation of SQL strings to a runtime library repository, which repository can access the database directly or through a database driver, such as a JDBC driver, without the need to imbed specific database accessing mechanisms in the application code.
2. A system according to claim 1, having a concurrent parameter setting mechanism wherein the runtime library repository can be set to access a particular data source and to generate data source specific database calls and SQL strings.
3. A object language software program that can generate application programming code from database maps and thereby provide a programming application, which delegates to a runtime library repository both of the functions of accessing a database and generating SQL strings that are specific to a database or to a JDBC driver for a type of database.

4. An object language software program according to claim 3, wherein the runtime library repository can be modified and tailored to optimize database access calls and to optimize the generation of SQL strings for a particular database.

5. A computer system according to claim 1, wherein the mapping system portion is designed to provide different maps for particular objects to different users and does not permit direct access of computer system users to JDBC drivers for any mapped data sources of the computer system, and wherein this mapping system provides varying levels of access to the mapped data sources for at least two different users of the same system, whereby a user only has access to a particular list of maps that are available to the security level of that user.

6. A computer system according to claim 1, wherein the mapping system portion is designed to provide to a system user who is accessing, creating or updating maps, or accessing objects on a system to make data changes related to a particular object and to promulgate the changes to that object as either local or global changes on the computer system.

7. A computer system according to claim 6, wherein the mapping system

portion provides an interface permitting an authorized user to edit or create the tables, fields, or attributes of a data map for an object as a table format or XML file format without requiring the user to have extensive knowledge of a particular relational database as a source of the data, or extensive knowledge about how to directly access that relational database.

8. A computer system according to claim 1, wherein the mapping system provides an interface and features that permit a user to access, create, or update the metadata of a map as a dynamic computer system update, without requiring the user to either open a new connection to the data source or to restart an object application program that is running while the user is dynamically evaluating or changing metadata for a map, and wherein the metadata of a map that a user can dynamically evaluate or change includes a map description of data or relationships between data, and wherein such map description is at least one member selected from the group consisting of a data source relationship, a relationship between at least two objects of an object application, and both a data source relationship and a relationship between at least two objects of an object application.

9. A local or distributed computer system comprising a fully synchronized caching system that synchronizes the caching of a delegated data source access

management object to relational mapping layer and the transaction coordination facilities of an application server, and the computer system comprises (i) a first data source referred to as the primary data source, (ii) a second data source referred to as the cache data source that is associated with an object to relational mapping layer to provide a data source cache for object applications, and (iii) a server having a transaction coordinator with the ability to register multiple data sources,

wherein:

(a) both the primary data source and the cache data source are registered with the transaction coordinator facilities of the server, and

(b) the cache data source acts as secondary data source to speed up data accesses for an object application and the cache data is plugged into the object to relational mapping layer, and

(c) registration of the cache data source with the transaction monitor of the server provides the feature that any changes to the cache will automatically be synchronized with the primary data source or record upon transaction completion, including commit or roll-back of changes for both data sources.

10. The computer system of claim 9, further comprising a mapping system

wherein object programming applications are tailored to delegate both the accessing of a data source and the generation of SQL strings to a runtime library repository, which repository can access the database directly or through a database driver, such as a JDBC driver, without the need to imbed specific database accessing mechanisms in the application code, and wherein the transaction monitor of the server calls the mapping system and in the call delegates accessing or updating of data in the data sources registered with the transaction monitor, and wherein the transaction monitor does not complete a transaction until it is notified by the mapping system that a data source has been updated by the mapping system such that any changes to the cache will automatically be synchronized with the primary data source or record upon transaction completion, including commit or roll-back of changes for both data sources.

11. The computer system of claim 10, wherein at least one cache data source can be set up as a memory-resident entire database, as a disk resident database, or as both a memory resident database and disk resident and in each case the cache data source is synchronized with the transaction monitor of the server.

12. The computer system of claim 11, wherein the at least one cache data source is a memory resident database that is a portion of a disk resident database and the

size of the memory allocated for the memory resident database can be set by the user of the computer system, or can be automatically allocated by the system according to pre-set protocols.

13. The computer system of claim 11, wherein the at least one cache data source is adapted to cache at least one member selected from the group consisting of data from a data source, a source structure map, an object or object definition, metadata, object schema, an object schema, and an XML file which defines or describes a member selected from the group consisting of data from a data source, a source structure map, an object or object definition, metadata, and an object schema.

14. A data source access repository that is a collection of information repositories and software program libraries, comprising at least one database access runtime library repository of software program libraries and system information, wherein

the data source access repository has the ability to make database specific calls and to generate database specific SQL strings by accessing at least one other software program library, information and logic to permit the data source access

repository to map information related to one or more of a member selected from the group consisting of relational databases, object databases and object programming schemas, and the like, and wherein

the data source access repository can access map information and use that information to make database specific calls and to generate database specific SQL strings that can be passed to a JDBC driver to access, retrieve and write data to and from a selected data source.

15. A data source access repository according to claim 14, comprising one or more items selected from the group consisting of maps of database schemas, object definitions, other programming definitions or parameters, object metadata, database settings, and complex relationship definitions.